BIOSTATS 640 – Intermediate Biostatistics
Spring 2022

Introduction to R
03 – Working Directory, R Markdown and Data Inspection

R in practice! In this lesson, we begin to develop some R work habits, three in particular and all beginning with the word "always": (1) set your working directory; (2) use R Mardown to produce transparent and reproducible work; and (3) inspect your data.

__1. Introduction to the HERS dataset
__2. Highlights from Lesson 02 – R Script, Packages and Data Description
__3. Set Your Working Directory
__4. Introduction to R Markdown
__5. Your First R Markdown: Data Inspection

Before you begin
If you have not already done so, install the package **stargazer**

#1. Introduction to the HERS Dataset

Source: Hulley S, Grady D, Bush T, Furberg C, Herrington D, Riggs B and Vittinghoff E (1998). Randomized trial of estrogren plus progestin for secondary prevention of heart diessase in postmenopausal women. The Heart and Estrogen/progestin Replacement Study. *Journal of the American Medical Association*, **280**(7), 605-613.

In the HERS study, Hulley et al. (1998) sought to determine if exercise, a modifiable behavior, might lower the risk of diabetes in non-diabetic women who were at risk of developing the disease. The question is a complex one because there are many risk factors for diabetes. Moreover, the type of woman who chooses to exercise may be related in other ways to risk of diabetes, apart from the fact of her exercise habit. For example, women who exercise regularly are typically younger and have lower body mass index (BMI); these characteristics also confer a risk benefit with respect to diabetes. Finally, the benefit of exercise may be mediated through a reduction of body mass index. Vittinghoff, Glidden, Shiboski and McCullogh (2005) consider portions of this data in their 2005 text, *Regression Methods in Biostatistics: Linear.Logistic, Survival and Repeated Measures Models* (Springer). Their dataset has n=2,763 observations on 37 variables.

In this lesson (03), we will explore a random sample of 1000 observations on eight (8) variables in the HERS dataset.

Data Dictionary

Position	Variable	Variable Label	Type	Codes	Missing data
1	HT	Hormone Therapy	numeric	0 = placebo 1 = hormone therapy	None
2	age	Age in years	numeric	Range: [44, 79]	None
3	raceth	Race/ethnicity	character	"1" = White "2" = African American "3" = Other	None
4	exercise	Exercise at least 3x/week	numeric	0 = no 1 = yes	None
5	diabetes	Diabetes	numeric	0 = no 1 = yes	None
6	BMI	Body Mass Index, kg/m^2	numeric	Range: [15.49, 49.51]	Yes
7	glucose	Glucose, mg/dl	numeric	Range: [67, 294]	None
8	LDL	LDL Cholesteraol, mg/dl	numeric	Range: [36.8, 365.2]	Yes

#2. Highlights from Lesson 02 R Script, Packages, and Simple Data Description

1. **Do all your work in either R Script (lesson 02) or R Markdown (this lesson)**

2. **R Script = [text editor] + [utility to send commands to console]**

3. **Tips: (1) Save often; and (2) Comment (these begin with #) extensively**

4. **To import Excel ".xlsx": File > Import Dataset > From Excel**
   Note: To import Excel ".csv": File > Import Dataset > From Text(readr)

5. **For basic descriptives: Use summary(dataframename) or summary(dataframename$variablename)**
   No package required.

6. **One (of many packages) package for numerical summaries is the package is {summarytools}**
   Functions illustrated were: freq( ), ctable( ), and descr( )

7. **A great package for data visualization is the package {ggplot2}**
   Graphs illustrated were: bar, side-by-side bar, histogram, and side-by-side histogram

#3. Set Your Working Directory
setwd( ) and getwd( )

Before you begin - Video
Consider watching this video.
(Source: MarinStats Lectures – R Tutorials) Setting up Your Working Directory (video, 8:25)

R keeps track of where to find things and where to put things – Working Directory
You're probably already familiar with the notion of folders on your computer that contain related things (files, graphs, docs, etc.) Similarly, in every R session, R connects to a *working directory*. This is a folder on your computer (or in the cloud, depending) where files will be read from and written to.

The default working directory depends on the operating system:

- Windows Users:
  My Documents

- Linux Users:
  /home/<**yourusername**>

- Mac Users:
  /Users/<**yourusername**>

getwd( ) to show the current working directory
Note: the parentheses are empty

There are 3 ways to set your working directoy

1. R Studio menu selection
2. **setwd("yourpathname")**
3. **setwd(file.choose( ))**

Good to know: Forward slashes versus backward slashes

Snag: On Windows systems, folders are separated by backslashes (e.g., My Documents\BIOSTATS 640\homeworks). The snag is that R treats a single backslash as a special code, resulting in two distinct uses of the single backslash.

The correct use of slashes in R depends on your operating system:

WINDOWS User, two choices:

- two single back slashes (e.g., My Documents\\BIOSTATS 640\\homeworks); or
- one single forward slash (e.g., My Documents/BIOSTATS 640/homeworks)

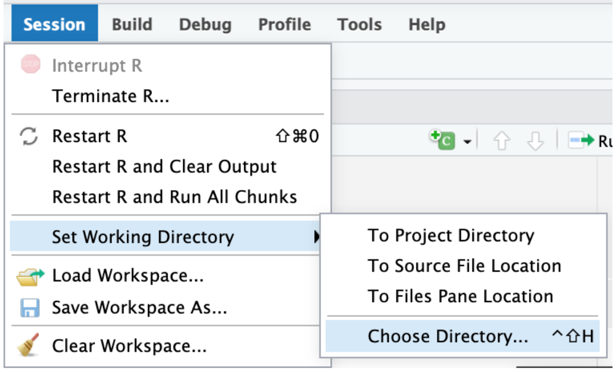
MAC User, one choice:

- one single forward slasch (e.g., ~Desktop/BIOSTATS 640/homeworks

Tip. Make your life simple. Use forward slashes all the time.

How to Set Working Directory: 1. Using R Studio Menus

From the top menu bar, click Session > Set Working Directory > Choose Directory
Browse to navigate to your desired folder. Click **CHOOSE.**



How to Set Working Directory: 2. Using setwd("yourpathname")

Don't forget. The path name must be enclosed in quotes.
Example (Windows): `setwd("My Documents/BIOSTATS 640/homeworks")`
Example (Mac): `setwd("~Desktop/BIOSTATS 640/homeworks")`

How to Set Working Directory: 3. Using setwd(file.choose( ))

`file.choose( )` also has nothing in the parentheses.
Execution of `file.choose( )` opens a window where you can browse to the folder of your choice.
The path is then shown in the console panel.
Do a EDIT/COPY/PASTE to complete your `setwd( )`

Hack. `file.choose( )` is very useful when pathnames have long names and perhaps embedded spaces and you don't want to have to type it out longhand.

#4. Introduction to R Markdown

Recall why R Markdown is so wonderful.
R Markdown produces a transparent and reproducible (permanent) record of your R code and comments and its output.

R Markdown in a nutshell.
R Markdown is formatting language utility. You use R Markdown to create documents that combine **text** and **executable R code**.

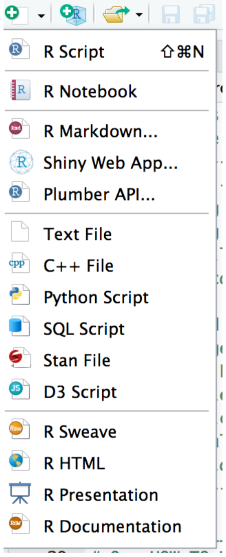
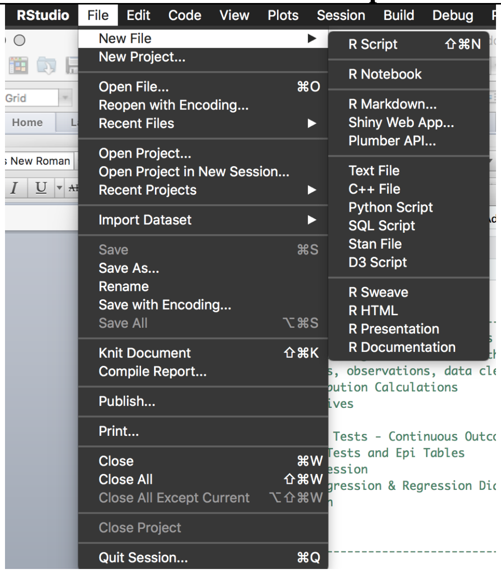
> **Regarding text**: Say good bye to MS Word
> You cannot "word process" the text in an R Markdown file as you would using MS Word.
> Instead you must use the text markup language provided.
>
> **Regarding executable R code**: Introduction to the R Code Chunk
> In an R Markdown file, R code is put into what are called R code chunks.
> R code chunks are conveniently shaded in gray
> Each begins with: ```{r your stuff, your options}
> And ends with: ```

When you execute an R Markdown file, this is called **knitting** or **rendering**. R Markdown includes a utility that allows you to produce output in several formats, among them:

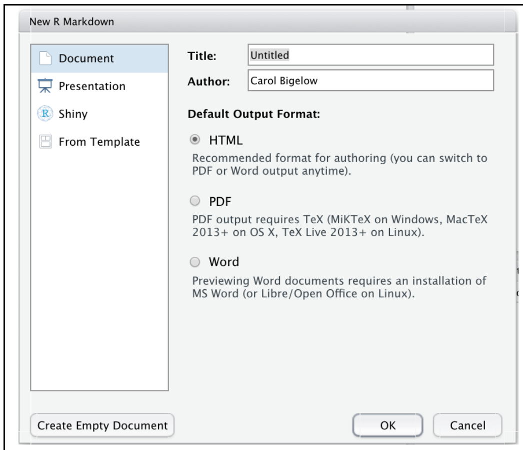
- html
- PDF
- Word
- and lots more

How to Create an R Markdown file:

From the source pane, click on either (+) or, from the toolbar, do FILE > NEW FILE > R MARKDOWN

The "+" symbol > R Script	FILE > NEW FILE > R Script
	

Make your choices



Title: Untitled

Author: Carol Bigelow

Default Output Format:

● HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

In the box Title: Enter a title of your choice

In the box Author: Edit (or not) as you like

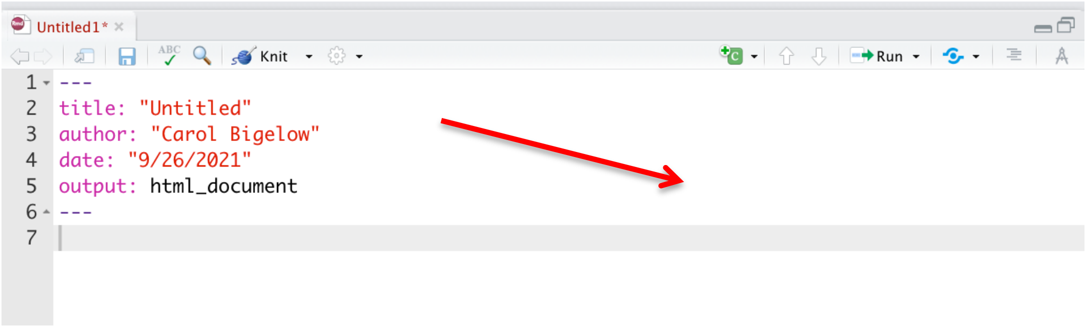
Under Default Output Format: Leave as is

At left, "Document" is highlighted: Leave as is

Click OK

Delete the pre-supplied R code chunks and text

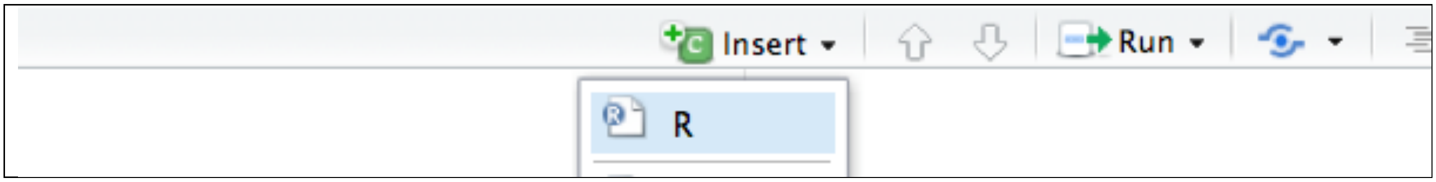
R returns a new R Markdown with some pre-supplied chunks and texts. These are here to help you understand chunks. In practice, you will delete the unwanted chunks and text.



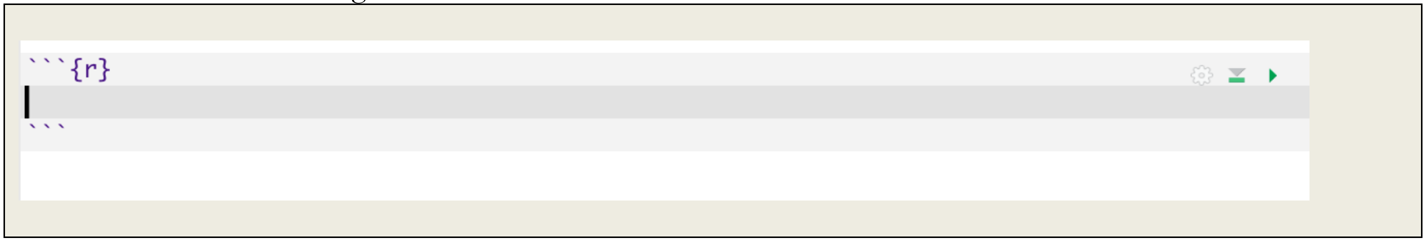
```
---
title: "Untitled"
author: "Carol Bigelow"
date: "9/26/2021"
output: html_document
---

```

How to Create a New Chunk

Click on the little green "insert a chunk" icon at top (on the right).
From the drop down menu, choose R



You should see the following blank chunk:

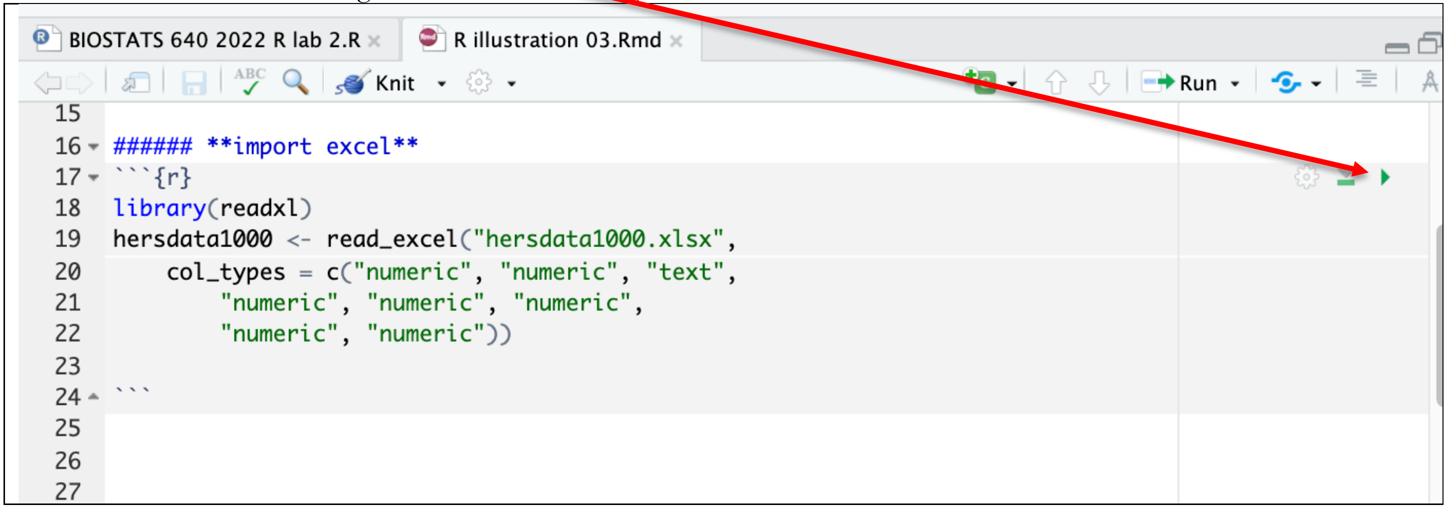


~~~
```{r}

```
~~~

How to Execute a Chunk

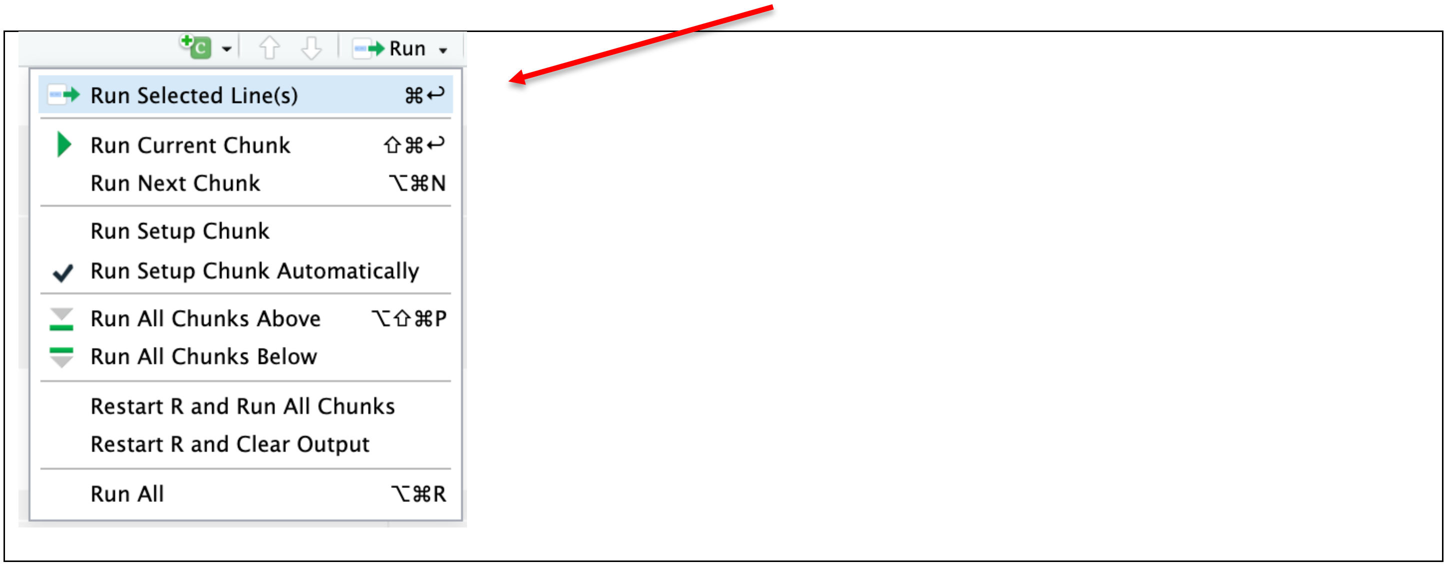
In a typical R session, you will create an R chunk, execute it, examine the output in your session, fix your mistakes, and then re-run your chunk. There are 2 ways to execute the current chunk:

Method #1. Click on the little green arrow at the far right



~~~
###### **import excel**
```{r}
library(readxl)
hersdata1000 <- read_excel("hersdata1000.xlsx",
    col_types = c("numeric", "numeric", "text",
        "numeric", "numeric", "numeric",
        "numeric", "numeric"))

```
~~~

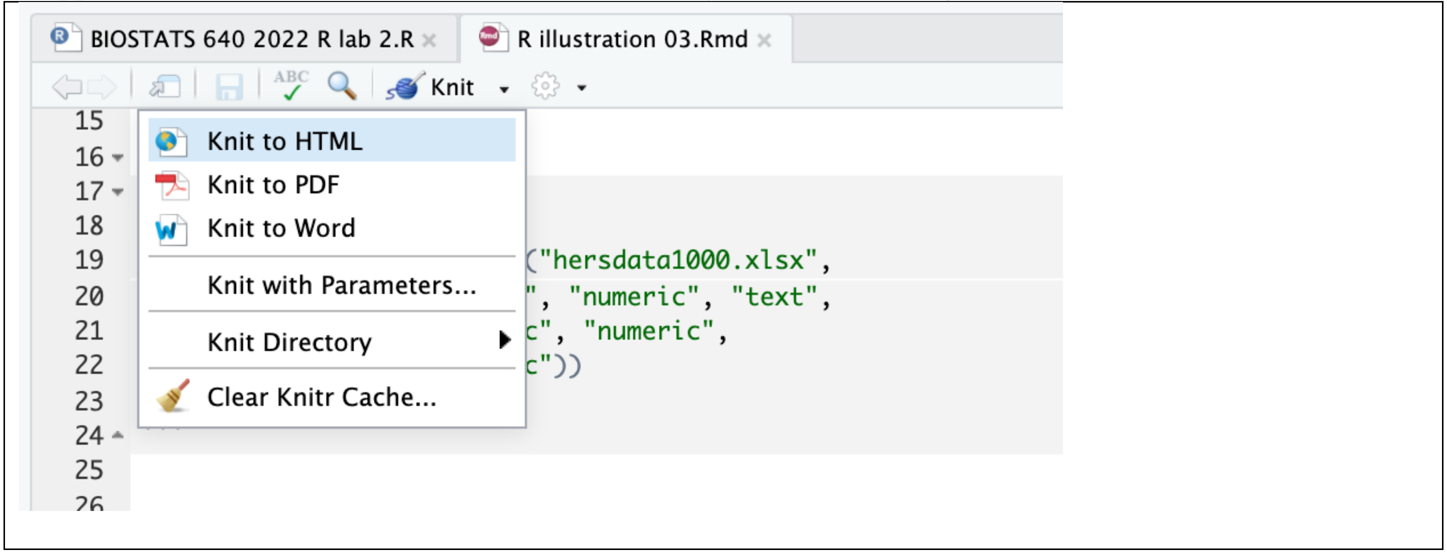
Method #2. Alternatively, you can use the drop down options under RUN at upper right



How to Render/Knit/Execute/Produce Your Report

Before you render, be sure to do a final save!

At upper left, click on the icon that looks like a ball of yarn with a knitting needle!



```
("hersdata1000.xlsx",
", "numeric", "text",
c", "numeric",
c"))
```

Good to Know. **You can also preview the looks of your report as you go along**

To do this, in the **`Environment pane`**, click on the tab **`Viewer.`** Now you can periodically knit to HTML; this will produce the output in your viewer for you to review.

#5. Your First R Markdown: Data Inspection

~~~
---
title: "R illustration 03"
author:  "Carol Bigelow"
date: "2/14/2022"
output: html_document
---


###### **initialize session**
```{r}
setwd("/cloud/project")                  # Set working directory
getwd()                                  # Check working directory
options(scipen=999)                      # Turn off scientific notation
rm(list = ls())                          # Clear the Decks
```


###### **import excel**
```{r}
library(readxl)
hersdata1000 <- read_excel("hersdata1000.xlsx",
    col_types = c("numeric", "numeric", "text",
        "numeric", "numeric", "numeric",
        "numeric", "numeric"))
```



###### **save as R dataset**
```{r}
save(hersdata1000,file="hersdata1000.Rdata")
```



###### **quick and easy summary**
```{r}
summary(hersdata1000)
```



###### **Inspection 1:  Check data types**
```{r}
str(hersdata1000)
```
~~~

~~~
###### **Inspection 2:  Create categorical variables (R calls these factors)**
```{r}

hersdata1000$racethf <- factor(hersdata1000$raceth)
hersdata1000$racethf <- factor(hersdata1000$raceth,
                               levels=c(1,2,3),
                               labels=c("White", "African-American", "Other"))

hersdata1000$HTf <- factor(hersdata1000$HT)
hersdata1000$HTf <- factor(hersdata1000$HTf,
                               levels=c(0,1),
                               labels=c("Placebo", "Hormone Therapy"))

hersdata1000$exercisef <- factor(hersdata1000$exercise)
hersdata1000$exercisef <- factor(hersdata1000$exercisef,
                               levels=c(0,1),
                               labels=c("No", "Yes"))

hersdata1000$diabetesf <- factor(hersdata1000$diabetes)
hersdata1000$diabetesf <- factor(hersdata1000$diabetesf,
                               levels=c(0,1),
                               labels=c("No", "Yes"))

str(hersdata1000)

```
~~~

~~~
###### **Inspection 3:  Assess Missingness**
```{r}
# Key:  Get count of missing values
# is.na( ) produces a 0/1 with 1=TRUE if observation is missing
# colSums( ) sums the 0/1 over all observations in that column
cat("\nNumber of missing values\nby Variable\n")
colSums(is.na(hersdata1000))

# Key:  Get count of complete observations
# !is.na( ) produces a 0/1 with 1=TRUE if observation is complete
# colSums( ) sums the 0/1 over all observations in that column
cat("\nNumber of complete observations\nby Variable\n")
colSums(!is.na(hersdata1000))

```

###### **Inspection 5: Assess Ranges**
```{r message=FALSE}
library(stargazer)

hersdata1000 <- as.data.frame(hersdata1000)
stargazer(hersdata1000,
          type="text",
          summary.stat=c("n","min","max"),
          title="HERS Data (random sample of 1000)")
```
~~~

~~~
######  **Inspection 5: Nice inspection of every variable**
```{r message=FALSE}
library(summarytools)

# NOTE:  The following will ONLY KNIT to HTML

print(dfSummary(hersdata1000), method='render')
```
~~~